RhodeCode CE/EE - Feature #5536

Ability to disable server-side SSH key generation

11.02.2019 19:42 - Catalin Salgau

Status:	Resolved	**Start date:**	11.02.2019
Priority:	Low	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	v4.17	**Spent time:**	0.00 hour
Sorting:		**Commit Number:**	

Description

Generating a private/public SSH key pair may be user-friendly, but shipping this from the server can pose security concerns.
Please add a way to disable key generation and/or provide client client-side key generation(Web Crypto APIs are available in most modern browsers)
Thanks

History

#1 - 13.02.2019 14:02 - Marcin Kuzminski [staff]

Hi,

What would be the reason to disable it?
In case an end-client doesn't trust he should generate his own set of keys.
RhodeCode is meant to be used in behind-the-firewall installations where users have total control over the server, and things that come from it should be trusted.

In the same way server generates some random passwords or access tokens.

#2 - 13.02.2019 15:52 - Catalin Salgau

Hi,
My reasoning for this request boils down to

- some consider private keys should never be outside of the owner's direct control. Offering server-side generation undermines policies/hardware token efficiency. Leaving this to a user's trust isn't always the preferable option.
- as it stands, key generation parameters are hard-coded in RhodeCode. Enforcing other parameters cannot be done without changing deployed source code. Keyword enforcing.

I would also add that

- from a key generation perspective, since RhodeCode is deployed as a prebuilt package with bundled dependencies, directly updating to account for weaknesses is not feasible in case of security concerns (pycrypto has been unmaintained for 5+ years. there is at least one CVE-registered weakness against the project since then)
- in case of compromise, all keys generated on the server would have to be considered leaked and must be invalidated, something complicated by key reuse by users (which should be safe for PKI systems)

I would also counter your last statement. Random tokens are not one-way functions. There is a much clearer understanding of password leakage/reuse (compared to that of keys).

#3 - 13.02.2019 16:12 - Marcin Kuzminski [staff]

- Target version set to v4.17

- Priority changed from Normal to Low

I agree with your statement about the end users should generate their own keys. The reality is that lots of users aren't fully aware of how to generate SSH keys and most importantly how to generate them securely.
This is why we expose an option to generate it on the server side, it's a handy feature to means to help users, and we try to follow best at the time practices to generate the keys in a secure fashion.

Those keys are meant to protect communication with the server they are generated on, if the server gets compromised it's basically game over and the key generation doesn't matter if someone compromised the server which those keys meant to protect communication with.

Finally to say: I think we should add a .ini flag to block the generation of keys as an *enforcement* mechanism, but it wouldn't be a priority feature to add for us.

We'll happily take that contribution as all of this is a part of the CE codebase.

#4 - 13.02.2019 21:07 - Redmine Integration

pullrequest created by csalgau (status: under_review).
https://code.rhodecode.com/rhodecode-enterprise-ce/pull-request/2275

#5 - 25.02.2019 21:52 - Marcin Kuzminski [staff]

- Status changed from New to Resolved

Thanks for your contribution, it's now a part of 4.16 release.

Some note on pycrypto, we updated sshpubkeys library to the latest version which doesn't use pycrypto anymore!

+1 on security :)

closing this ticket now.

#6 - 28.02.2019 13:52 - Redmine Integration

Commit 6cd9b76816a5 by csalgau csalgau@users.sourceforge.net on default branch changed this issue.
https://code.rhodecode.com/rhodecode-enterprise-ce/changeset/6cd9b76816a5193de8c8b2018193e239ee7fc276